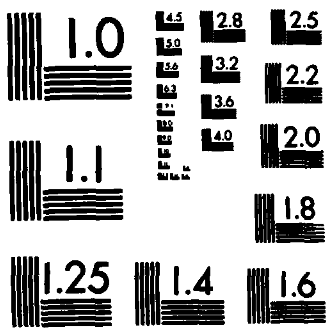
AD-A122 917 CABLE AVOIDANCE STUDY(U) MAGNAVOX GOVERNMENT AND INDUSTRIAL ELECTRONICS CO MAHWAH NJ ELECTRO-OPTICAL SYSTEMS R HERMES 29 OCT 82 EOSR-839 DAAK70-81-C-0177 1/1

UNCLASSIFIED F/G 1/4 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Report No. EOSR-839

12

AD A122917

FINAL TECHNICAL REPORT

CABLE AVOIDANCE STUDY

Author: Dr. Robert Hermes

EOSR No. 839

October 29, 1982

Magnavox
Magnavox Government and Industrial Electronics Company
ELECTRO-OPTICAL SYSTEMS
46 INDUSTRIAL AVENUE, MAHWAH, N.J. 07430 • TEL: 201-529-1700 • TWX: 710-988-5672

Prepared for:
Department of the Army
Night Vision & Electro-Optics Laboratory
Fort Belvoir, Va. 22060

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER EOSR 839	2. GOVT ACCESSION NO. AD-A122917	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Final Technical Report, Cable Avoidance Study		5. TYPE OF REPORT & PERIOD COVERED Final 23 Sept. 81 - 29 Oct. 82
		6. PERFORMING ORG. REPORT NUMBER EOSR No. 839
7. AUTHOR(s) Dr. Robert Hermes		8. CONTRACT OR GRANT NUMBER(s) DAAK70-81-C-0177
9. PERFORMING ORGANIZATION NAME AND ADDRESS Magnavox Government & Industrial Electronics Co., Electro-Optical Systems 46 Industrial Avenue, Mahwah, N.J. 07430		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Army Night Vision & Electro-Optical Laboratory Fort Belvoir, Virginia 22060		12. REPORT DATE 10/29/82
		13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Infrared
FLIR
Cable Avoidance
Real Time
Semi-Local Line Discriminator
Nearest-Neighbor Correlation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The problem of real-time wire extraction from video scenes for helicopter navigation is addressed. A combination of "semi-linear" local line detectors and "semi-local" line discriminators is developed to optimize detection and false alarm rejection. While the results are quite good, further software development is indicated to improve both detection and noise rejection for pilot displays. Two directions for further development are suggested. A video tape demonstration of the current algorithm applied to scenes containing cables accompanies this report.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Summary

This report along with a video demonstration tape documents the efforts expended in investigations conducted at Magnavox into the development of real-time algorithm techniques which can be used in FLIR imagery for the detection of cables in the field of view.

A combination of "semi-linear" local line detectors and "semi-local" line discriminators were developed to optimize detection and false alarm rejection.

Conclusions and recommendations for future efforts in this area are discussed in Section 6.

This program was monitored by Mr. Jamie Jones of the Night Vision & Electro Optics Laboratory, Fort Belvoir, Virginia 22060.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

This final technical report covers the progress and efforts expended by Magnavox Government and Industrial Electronics Company (MAGIEC) towards the objectives of Contract DAAK70-81-C-0177 for the period of 1 October 1981 to 1 September 1982, and is submitted together with a video demonstration tape in compliance with Contract Data requirements List Item A001. This contract comprises an investigation into the real-time algorithm techniques which may be used in FLIR imagery for the detection of cables in the field of view. The video tape shows examples of the developed algorithm applied to video scenes containing cables.

1.1 BACKGROUND

It has long been a problem for pilots of low flying helicopters to detect and avoid wire obstacles such as telephone lines and power cables. Past experience has shown that wire strikes may occur during routine day operation. With increased emphasis on Nap-of-the-Earth (NOE) flight as a tactic to avoid detection by threat radars and optical systems, the wire avoidance problem becomes even more critical. In addition, NOE operations are now expanding into periods of limited visibility and darkness through the use of IR systems such as the Pilot's Night Vision System (PNVS) on the Advanced Attack Helicopter (AAH).

NOE flight under the best of conditions is a demanding pilot task. NOE flight at night with a PNVS creates an environment in which the pilot must work at near maximum capability. It is in this environment that the pilot is easily detracted by other imagery in the scene or other flying tasks and the ability to perceive faint wire obstacles is seriously degraded. Therefore, the problem addressed in this contract is to determine methods which will detect wire obstacles in the scene video of a PNVS.

Previous solutions to the wire avoidance problem have addressed the

general case of both day/night operations and have been independent of other on-board sensors. These solutions have required the addition of dedicated sensors to the helicopter resulting in greatly increased weight, complexity, and cost. In addition, the effectiveness of these sensors has been marginal. For the AAH, however, the PNVS represents a high performance FLIR which through more effective utilization can provide imaging of wire obstacles. The solution investigated here is the extraction of wire obstacles from video imagery using advanced image processing techniques. With the development of the common module Digital Scan Converter (DSC), most of the required hardware will already be on-board the helicopter. This approach also has the advantage of being passive whereas the previous solutions have required active systems, thus increasing the helicopter's vulnerability in a combat environment.

2.0 THE PROBLEM

A capability for cable avoidance through passive IR image processing techniques requires the ability to discern thin lines in the presence of competing scene structure, fixed or moving pattern noise, and random noise, based solely on the brightness distribution measured over the scene. This is a difficult requirement under the best of circumstances, and necessitates an algorithm sensitive to those characteristics in the brightness distribution which are unique to lines.

Such an algorithm must be able to distinguish between lines and scene clutter such as edges, points, clusters, and similar features (intensity variations) which arise from buildings, roads, foliage, etc., found in any terrain scene. These features can effectively hide or break up the unobtrusive signatures of hanging wires.

An even more diffucult task is for the algorithm to distinguish between lines resulting from hanging cables and lines due to other phenomena. The most serious problems arise from streaking caused by detector non-uniformities and from raster crawl caused by incompatibilities in FLIR and TV scanning modes. However, the advent of the digital scan converter technology, with its TV compatible output and its gain and level equalization preprocessing will considerably reduce these problems. The algorithm should be able to discriminate against any residual effects. Lines due to other scene components besides cables may not be so easily dealt with, but on the other hand, are not that severe a problem. Nevertheless, the ideal algorithm should be able to distinguish between these lines.

Random system noise and high spatial frequency scene fluctuations due to foliage, etc., has small line segment structure and point structure. The line segment structure results in algorithm false alarms, while point structure

near true lines alters the line signatures and results in algorithm failure to detect real signals. Digitization noise, and signal aliasing effects of the sampling process intrinsic to digitizing a scene also have a corrupting effect on line signatures. A good algorithm must be able to operate effectively in poor signal-to-noise ratio environments.

The problem of cable avoidance from an algorithm point of view can be separated into two distinct tasks: (1) detecting potential line (cable) segments in the scene in the presence of noise and clutter and (2) connecting the proper detected line segments into extended lines further eliminating false alarms. The first task requires a local algorithm utilizing an operator that is sensitive to the scene intensity statistics in the neighborhood of a given scene element, or pixel*. Its difficulty is in discriminating between a line element and a non-line element (e.g., an edge element or point element). An overly restrictive algorithm will discard too many real line segments and an overly loose algorithm will keep too many clutter elements (false alarms). The second task requires a global algorithm with the capability for remembering all detected line segments from across the entire scene, discriminating between potential cable segments and false alarms, and connecting the cable segments into extended lines which are true representations of the original cables. This algorithm may require bridging large gaps caused by intervening scene objects.

There are two constraints that make the problem even more difficult. First, the complete algorithm must operate in nearly real time, i.e., in a time short enough to follow scene movement incurred due to vehicle or FLIR motion, and in any case short enough to allow the pilot to take evasive action. Second, the algorithm must be implementable in a small, light weight format

*A picture element is completely determined by its position coordinates and its brightness value.

(microprocessor, etc.) for inclusion in the on-board electronics.

Due to the limited scope of this effort it was decided to confine the investigation to the local detection algorithm since this appears to be the critical phase of the problem. It may be possible to allow the eye to connect line segments that have been sufficiently enhanced to be immediately obvious to the pilot. Furthermore, failures of the detection algorithm obviates the need for a connection algorithm. However, some general concepts and aspects of a connection algorithm will be discussed in terms of the future directions the algorithm should take for a successful completion of the task.

3.0 THE CANDIDATE ALGORITHMS FOR LOCAL DETECTION

Any algorithm that detects lines must be based on some essential characteristic or group of characteristics of a line that make it different from other picture elements. The essential ingredient of a linear feature is that it have continuity in one direction, but represent some deviation from the norm in any other direction. For lines, it is the picture intensity that is continuous and slowly varying along the line (i.e., on and in the direction of the line), but has an extremum in any other direction. Thus, a line has three key qualities:

1) A direction

2) A finite extent in that direction

3) An extreme intensity value across that direction. (For real lines of finite width the intensity gradient should be a maximum along the perpendicular to the line's direction.) The candidate detection algorithms will make use of some or all of these qualities.

3.1 MEDIAN FILTERING

One algorithm which makes use of a non-unique statistical feature of a line applies the compliment of a median filter to the scene. Since the elements of a line are greater (or less) than their off line neighbors, they will never be the median value in their neighborhood, and in fact these line elements make a maximum difference with the neighborhood median. Consequently, if we subtract the neighborhood median from each pixel, we should get extrema on lines and a washed out scene away from lines. However, while this technique can discriminate between lines and edges, it cannot do so between lines and points since point elements by definition are greater (or less) than their neighbors. Because this algorithm fails to take into account the directivity and connectedness of lines vis a vis points, it enhances points as well as lines.

The resulting "line-enhanced" scene is much noisier than the original scene. Although originally it was planned to pursue this technique based on our previous observations of its line enhancing capability, it has been discarded because of its noise enhancing side effect. Instead a search was undertaken for a technique which makes use of three key qualities of a line mentioned above.

3.2 LOCAL DETECTOR ALGORITHM STRUCTURE

A line element has a unique relationship to its neighbors (i.e., to other nearby line elements and to neably off-line elements) determined by those qualities which define a line. To detect a line element we must first determine the value of the neighboring elements or at least sample the neighborhood, and then verify that the proper relationship exists for some choice of direction (i.e., for some assignment of on-line and off-line elements). This is accomplished by choosing a set of sampling templates, each member of which assumes a different orientation of the line and assigns the identity of the on-line samples and the off-line samples which will be tested, based on the assumed direction. If each member sampling template is applied to a picture element and its neighborhood, the relationship of the picture element to the neighborhood samples will conform to the line detection criteria only when the picture element is indeed a line element and then only when the assumed direction is correct. The algorithm can then be structured as follows:

a. Apply the sampling template set to each element and its neighborhood.

b. Apply detection criteria for evidence of a line for each template.

c. Compare evidence for each direction and choose direction which gives "best" evidence for a line, if one exists.

d. If no evidence exists, or evidence is not sufficient (i.e., below threshold) assume no detection.

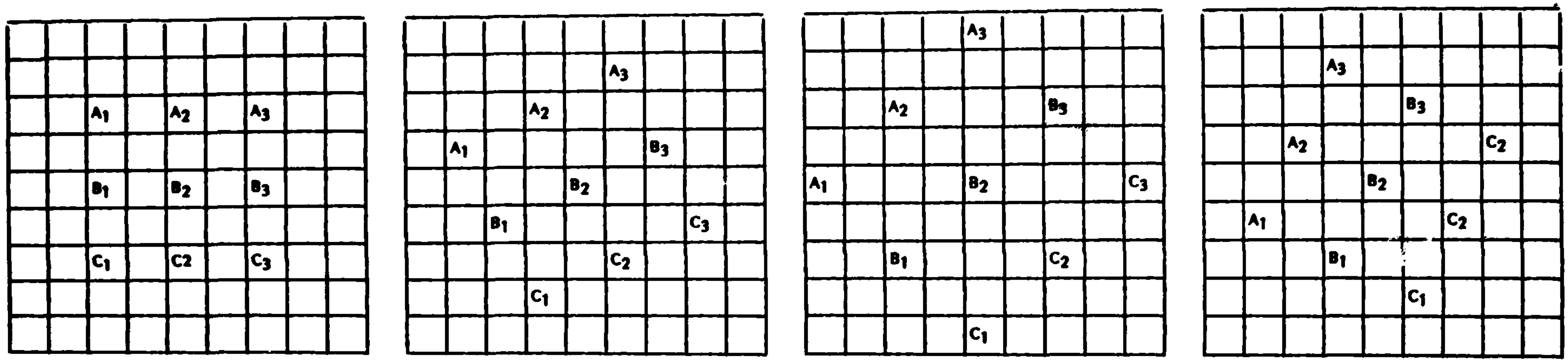
3.2.1 The Sampling Template Set

The size and sampling configuration of the sampling template set are important in determining the direction and width of a line. A 3 x 3 sampling template set allows only four (4) different directions giving a resolution of 45°. A 5 x 5 set allows for eight (8) possible directions (a resolution of 22.5°) although the sampling configuration is somewhat awkward for some directions. Figure 3.1 is an example of a 3 x 3 neighborhood sampling template for which the assumed line direction is vertical. B_2 is the queried pixel, B_1 and B_3 are its neighboring on-line pixels, and the A's and C's are the parallel off-line neighbors.

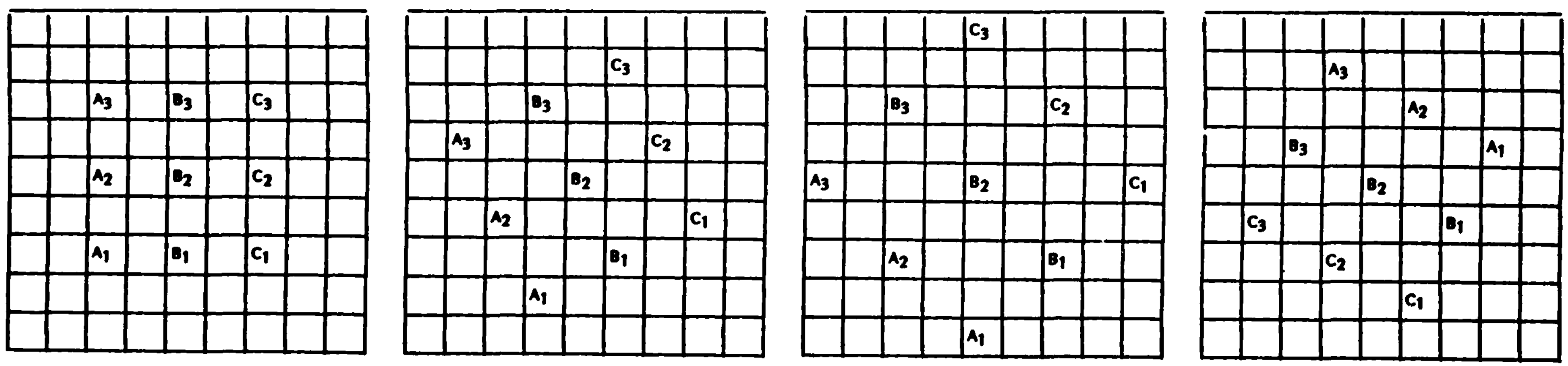
A_1	B_1	C_1
A_2	B_2	C_2
A_3	B_3	C_3

Figure 3.1

Both the 3 x 3 and 5 x 5 sets require that the samples be contiguous. This means that only thin lines (approximately 1 pixel wide) will be detected. If the samples were separated, thicker lines would also be detected. In order to have a template set with at least 8 possible orientations (22.5° resolution) and with a non-contiguous symmetric sampling configuration, a 9 x 9 neighborhood need be sampled. Therefore a 9 x 9 sampling template set was chosen for this investigation and is shown in Figure 3.2.



(a) k = 1; $\theta_1 = 0^\circ$ (b) k = 2; $\theta_2 = 22.5^\circ$ (c) k = 3; $\theta_3 = 45^\circ$ (d) k = 4; $\theta_4 = 67.5^\circ$



(e) k = 5; $\theta_5 = 90^\circ$ (f) k = 6; $\theta_6 = 112.5^\circ$ (g) k = 7; $\theta_7 = 135^\circ$ (h) k = 8; $\theta_8 = 157.5^\circ$

Figure 3.2 The 9 x 9 Sampling Template Set

3.2.2 LOCAL LINE DETECTION CRITERIA

To constitute a line detection the on-line sample elements must be greater (or less) than the neighboring off-line sample elements to either side. Mathematically, this can be formulated by computing the differences of the on-line sample elements minus the off-line sample elements (e.g., $B_i - A_i$, $B_i - C_i$) and requiring that they all have the same sign. However, there are various ways to formulate this criteria, each with a somewhat differing result. These approaches can be classified as linear, semi-linear, and non-linear. They differ only in the degree to which they employ the unique qualities of a line.

3.2.2.1 The Linear Algorithm

The linear approach requires that the average of the on-line samples be greater (or less) than the average of all the off-line samples by some threshold amount or no detection is encountered. If $a = \Sigma A_i$, $b = \Sigma B_i$, and $c = \Sigma C_i$ (ref. Figure 3.2) and if E is the evidence for a line then the linear algorithm can be simply stated mathematicaly by the following equations,

$$E = 2b - a - c \text{ , if } |E| \geq t; \qquad (3.1)$$
$$E = 0 \text{ , otherwise.}$$

Due to the non-uniqueness of this statistical property, this method responds to edges and isolated points as well as to lines although its line response is much stronger. On the other hand, because it averages over all off-line elements at once, it tends to filter out the corruptive effect of noise on line segments. Nevertheless it was decided to eliminate the linear algorithm immediately, based on its inability to discriminate against edges and points.

3.2.2.2 The Non-Linear Algorithm

The non-linear algorithm requires that each on-line sample in-turn be greater (or less) than either of its two closest off-line sampled neighbors. An example of a non-linear line detector is,

$$E = 2b - a - c$$
$$\text{if } B_i - A_i \geq t \text{ and } B_i - C_i \geq t \text{ for all } i \qquad (3.2)$$
$$\text{or if } B_i - A_i \leq t \text{ and } B_i - C_i \leq t \text{ for all } i;$$
$$E = 0, \text{ otherwise}$$

This algorithm discriminates against both edges and isolated points, but it has a problem dealing with short gaps in a line caused by interfering features and (especially) corruptive noise. While this algorithm is effective in eliminating false alarms, it has a low detection capability for lines in the presence of noise. The above algorithm was coded and tested along with the semi-linear algorithm.

3.2.2.3 The Semi-Linear Algorithm

The semi-linear algorithm is a compromise between the linear and the non-linear algorithms. While it does not average over all off-line samples, neither does it treat each off-line sample separately. Rather, it averages separately over the two sets of off-line samples (one set to either side of the assumed line). That is, the semi-linear algorithm requires that the average of the on-line samples be greater (or less) than either off-line set-average by some threshold amount. An example of a semi-linear detection algorithm is given by the equations,

$$E = 2b - a - c$$
$$\text{if } b - a \geq t \text{ and } b - c \geq t$$
$$\text{or if } b - a \leq t \text{ and } b - c \leq t; \qquad (3.3)$$
$$E = 0 \text{ , otherwise.}$$

This semi-linear approach eliminates edges but is still somewhat responsive to isolated points, although less so than the linear algorithm. However, it is less influenced by corrupting noise near line elements. This algorithm was also coded and evaluated in this study.

3.2.2.4 A Variation

One can choose another estimate for the evidence besides the sum of differences given in equations 3.1, 3.2, and 3.3. For example, in the linear approach one could choose the product of the difference,

$$E = (b-a) \cdot (b-c) \quad \text{if } (b-a) \geq t \text{ and } (b-c) \geq t$$
$$\text{or if } (b-a) \leq t \text{ and } (b-c) \leq t; \qquad (3.4)$$
$$E = 0 \text{ , otherwise.}$$

This variation should improve the discrimination between lines and edges. This is because the product favors lines having similar off-line statistics to either side of the line, vis a vis, lines whose off-line samples to one side of the line are much stronger than the off-line samples to the other side of the line, a condition which strongly resembles an edge corrupted by noise.

3.2.3 MAGNITUDE AND DIRECTION ALGORITHMS

Once the evidence for a line has been computed for each orientation of the sampling set it then needs to be evaluated to determine which orientation has the strongest evidence. Since the gradient is greatest when traversing across the line perpendicularly, the evidence (differences) should be strongest when the chosen on-line elements are most closely aligned with the actual direction of the line. If we take as the direction of the line, the direction associated with the sampling template which has the strongest evidence, we should be able to evaluate the direction of the actual line to within $\pm$ 11.25°. At this point we can also define some measure of the strength of the line based on the evidence.

3.2.3.1 The Maximum Evidence Approach

The most straightforward way is to just choose the value of the maximum evidence. If VAL is the magnitude of the line, and THETA its direction (i.e., the angle it makes with the horizontal axis as shown in Figure 3.3), then

$$VAL = \max [E_k] = E_{km}, \tag{3.5}$$

where E_k is the evidence from the k^{th} sampling template and k_m is the index of the template having the maximum evidence. The angle associated with the k^{th} template is given by,

$$\theta_k = \frac{(k-1)\,\pi}{8} \text{ radians} \tag{3.4}$$

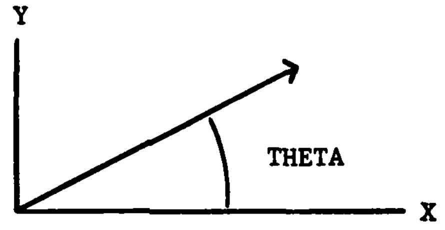


Figure 3.3

VAL can be thresholded to reduce false alarms. This algorithm was tested and evaluated in the study.

3.2.3.2 <u>The Statistical Approach</u>

Another more complex algorithm uses circular statistics to develop a more accurate estimate of the direction and strength of a line. The direction can be obtained from the weighted mean orientation $\bar{\theta}$ by

$$X = \Sigma\ E_k \cos (2\ \theta_k) \tag{3.5}$$

$$Y = \Sigma\ E_k \sin (2\ \theta_k) \tag{3.6}$$

$$\bar{\theta} = (1/2) \tan^{-1} (Y/X). \tag{3.7}$$

$$THETA = \theta_k \text{ nearest to } \bar{\theta}. \tag{3.8}$$

The strength of the line can be derived from the variance of the weighted distribution of directions from the mean direction. If the variance is given by,

$$\rho^2 = 1 - (X^2 + Y^2)/(\Sigma\ E_k)^2 \tag{3.9}$$

$$\text{then} \quad VAL = 1 - \rho^2 = (X^2 + Y^2)/\Sigma\ E_k)^2 \tag{3.10}$$

VAL is now a measure of the unanimity or agreement of the statistical directional data with the weighted mean orientation.

3.2.3.3 The Symmetry Parameter

In addition one can define a third parameter which can be used as a figure of merit of a line. This parameter depends on the symmetry of the line across the direction of the line, and is given by,

$$MU = (2b - a + c)/4 \tag{3.11}$$

$$(SIGMA)^2 = [2(b - MU)^2 + (a - MU)^2 + (C - MU)^2]/4 \tag{3.12}$$

$$KAPPA = (2b - a - c)/4 \tag{3.13}$$

and

$$SYM = \frac{KAPPA}{SIGMA} \tag{3.14}$$

where $0 \leq SYM \leq 1.$

SYM is a measure of the degree to which the off-line samples to either side of the line have the same value. As such this parameter helps discriminate against edge like features. Often times an edge which is corrupted by noise can meet the statistical requirements of a line, although its symmetry (as measured by SYM) would be low. SYM can also be thresholded to reduce false alarms. An algorithm using THETA, VAL, and SYM to evaluate the evidence for a line was used in this study.

4.0 THE SEMI-LOCAL LINE DISCRIMINATOR

In order to reduce the false alarm rate without overly degrading the detection capability, a discrimination technique is required that is independent of the detection mechanism. The previous algorithms made use of the local property of a line that the pixel value is an extremum on a line when traversing across the line and that the extremum is largest when the direction traversed is normal to the line direction (Property 3 of section 3.0). What has not been employed is Property 2 of section 3.0, namely, that a line has finite extent along its direction. That is, there is some continuity along the direction of the line in both magnitude and direction, particularly in direction since fluctuations in background (off-axis texture, etc.) could affect the magnitude. Therefore, a detected line segment can be expected to have on-line neighbors which are also detected and which have the same direction. To make this determination about a detected line segment requires extending the neighborhood of consideration, i.e., merging neighborhoods in the direction of the detected line segment. In this sense it is a semi-local algorithm, a compromise between the purely local detection algorithm and the global connection algorithm which would link locally deleted line sections into continuous lines which extend across the scene.

The semi-local line discriminator, examines a detected line element to see if has extent along its direction, accepting only those detected line elements which have detected line elements at either end of its direction, or at least its immediate neighborhood which line segments have the same or similar direction and perhaps the same sign. (The sign of a line element will be considered positive if the local on-line extremum is a maximum, and negative if it is a minimum. Having the same sign is a compromise between having the same magnitude and requiring no magnitude correlation whatever.) To make these

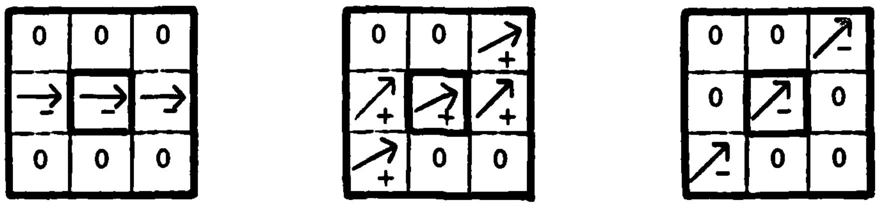
comparisons requires more memory since it becomes necessary to store the direction and magnitude (or just sign) information of the neighboring detected line elements for future comparisons. There are numerous variations possible in this correlation process. Two algorithms were chosen for this study, one based on nearest neighbors and one based on second nearest neighbors.

4.1 NEAREST-NEIGHBOR CORRELATION

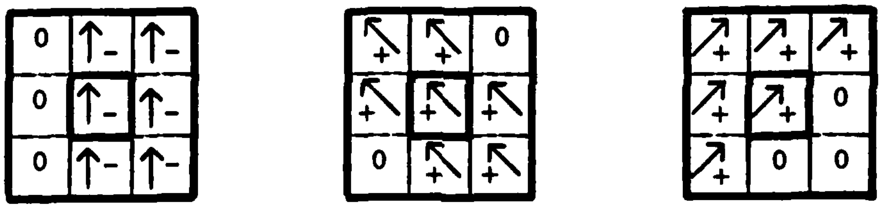
The first algorithm that was used tested only nearest neighbors for a correlation of direction and sign. For nearest neighbors there are only four orientations where a direct extension of the detected line element leads to an unambiguous choice of neighboring on-line elements (and their orientation). Since there are eight possible line orientations, the algorithm used did not require specific neighboring elements to be detected line elements of a given orientation and sign. Instead it was required that the neighborhood contain at least a specified number (N) of detected line elements of the same orientation and sign. To carry out the comparison, each detected line element was assigned a direction number between 1 and 8 and a sign, positive for a bright line and negative for a dark line. (For example, a bright vertical line is assigned a value of +5 whereas a dark line at 45° is assigned a value of −3.) If N is set to 2 then Figure 4.1 shows sample 3 X 3 neighborhoods of detected line elements for which the central element is considered a valid line element.

4.2 SECOND-NEAREST NEIGHBOR CORRELATION

If second nearest neighbors are used, the test is more stringent since it requires a larger extent of the continuity of the line segment. Furthermore, all eight possible line orientations lead to an unambiguous choice of which second nearest neighbors should also be line segments and what their orientations should be. An extrapolation of the central detected line element in a 5 x 5 neighborhood should intersect two second nearest neighbors. The



(a) line value: -1 | (b) line value: +2 | (c) line value: -3



(d) line value: -5 | (e) line value: +7 | (f) line value: +3

Figure 4.1. Some Sample 3x3 Neighborhoods Containing Valid Central Line Elements for N Equal to 2

algorithm used in this study then requires that the two intersected second-nearest neighbors also be detected line elements with the same orientation and sign as the central detected line element. No requirements are specified for the other nearest neighbors. Figure 4.2 shows sample second nearest neighbor configurations for which the central element is considered a valid line element.

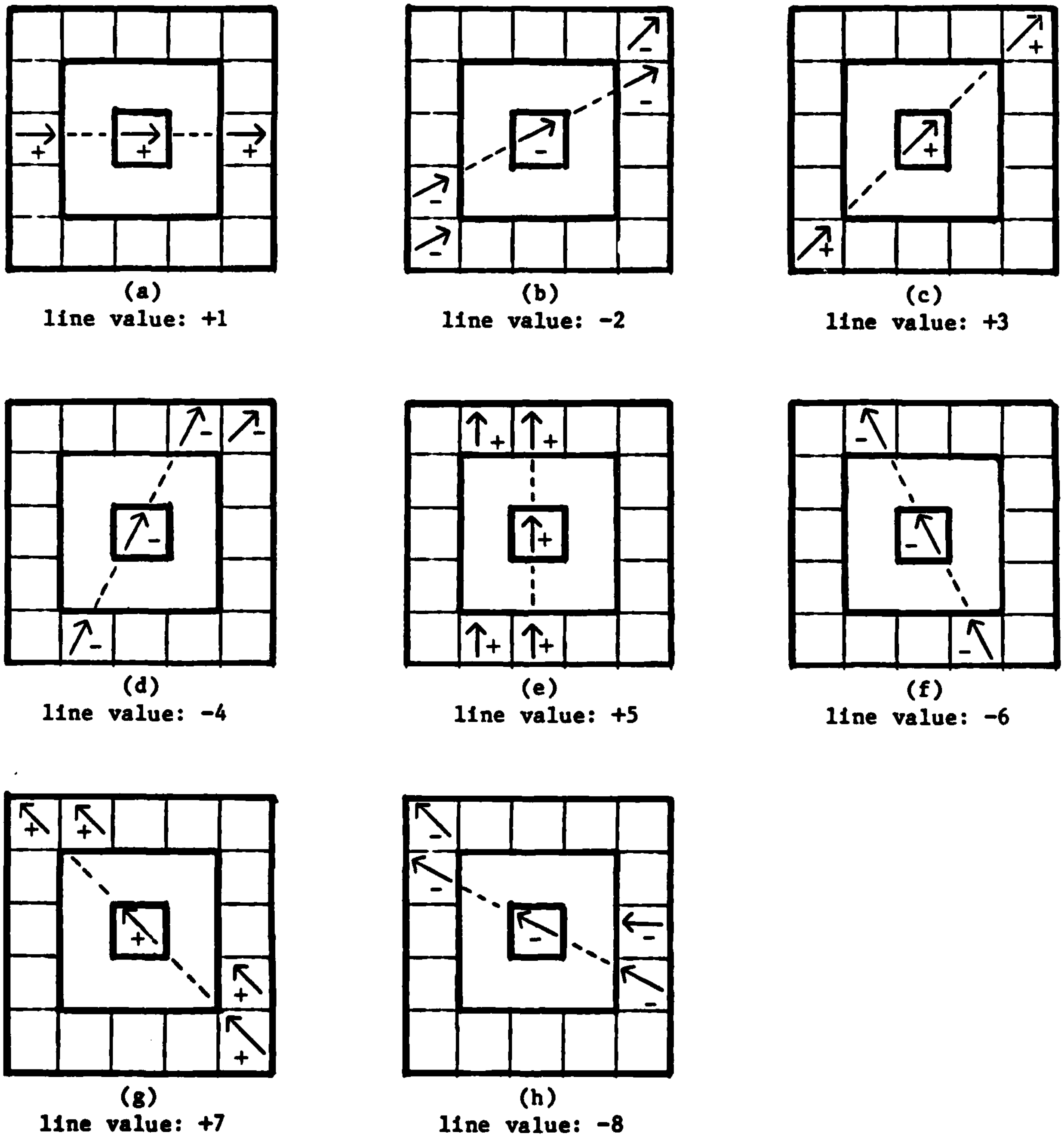
4.3 EVALUATION OF THE SEMI-LOCAL LINE DISCRIMINATOR

The algorithm using nearest neighbors was coded and tested in such a way that one could vary N, the minimum number of nearest neighbors required to be line elements of the same orientation and sign. This algorithm turned out to be quite good, yielding a significant improvement in the false alarm rejection. The best case was for N equal to 2, and this value was used in all subsequent test. However, there was still room for more improvement.

The algorithm using second nearest neighbors was also coded and tested. The result was a remarkable improvement in the rejection of false alarms, even better than the algorithm using only nearest neighbors.

Since the above two semi-local line discriminators were independent, they could effectively be joined to yield a combined algorithm that was better than either alone. This was tried and found to be the case. In fact the improvement over the local detector results were so dramatic that any improvements resulting from interchanging the various local detector algorithms was comparatively unimportant. The combined semi-local discrimination algorithm was tried with each of the potential local line detection algorithms and found to work approximately equally well with each. The key consideration in choosing the local detection algorithm is therefore the computation time. Choosing the fastest local algorithm, namely one which takes the evidence as a sum of on-line minus off-line differences instead of a product, and one which simply chooses

Figure 4.2 Sample Second Nearest Neighbor Configurations Yielding a Valid Central Line Element



the maximum evidence as indicating the line direction, should yield the best overall result. Since there is no significant difference in computation time between the non-linear and the semi-linear local detection algorithms, the preferred semi-linear algorithm was chosen.

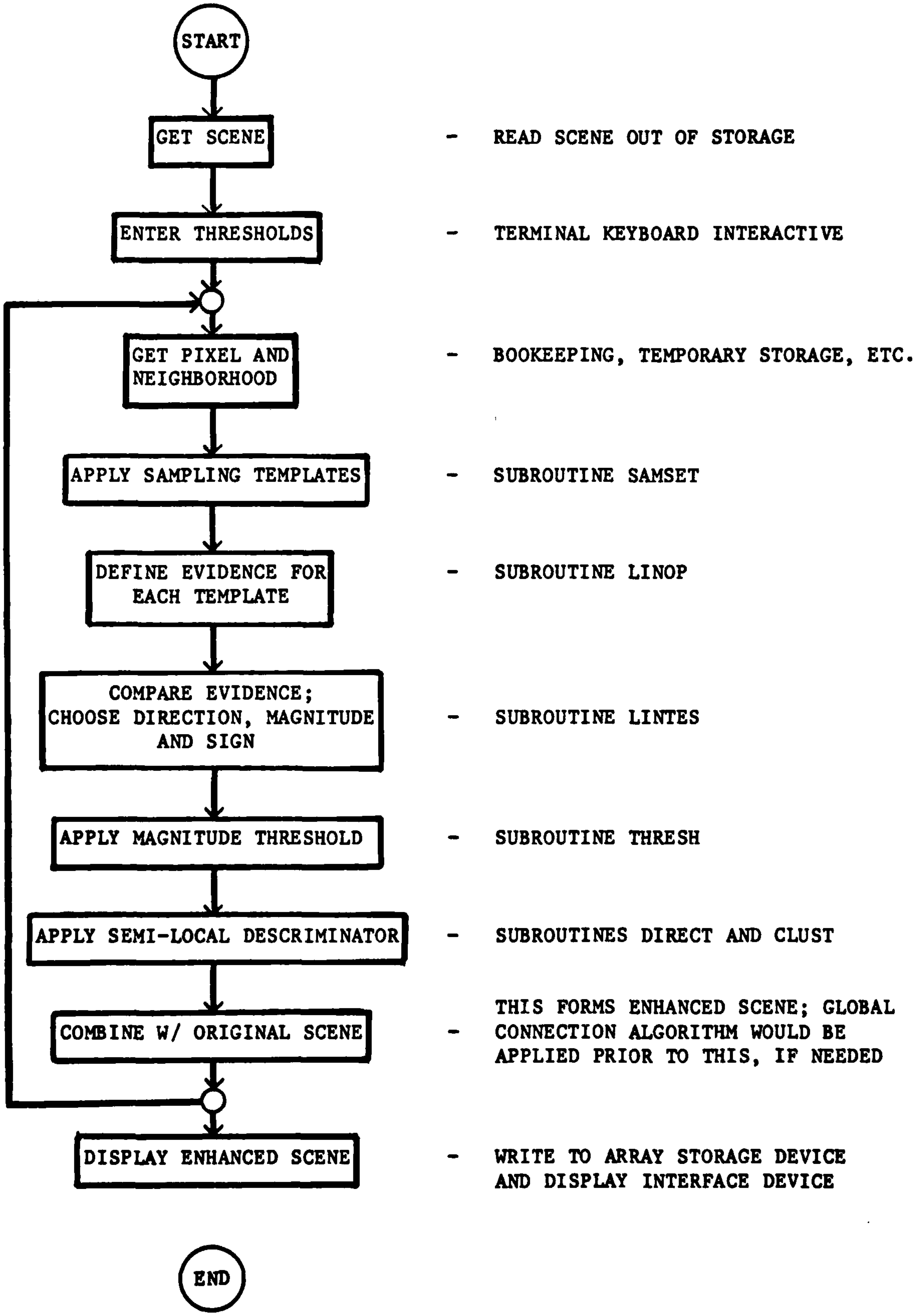
The possibility of combining the local detection and semi-local discrimination algorithms with the median filtering algorithms was considered. This was tested by first subtracting the median filtered scene from the original scene to give the "enhanced" scene. Then the algorithms evaluated above were applied to this scene. The results were appreciably degraded from the results without the median filter algorithm, due principally to the increased false alarms caused by the point enhancing property of the median filter algorithm.

One result of the improved discrimination of the semi-local algorithm is that the detection thresholds can be lowered substantially without a large increase in false alarms. The reduction in optimum threshold yields a more sensitive total algorithm. However, its sensitivity is still somewhat limited. What improvements can be made in sensitivity in subsequent efforts is discussed later.

5.0 FORMAT OF THE SOFTWARE IMPLEMENTATION

In order to establish a framework in which to compare the variations in the cable enhancement algorithm, it was decided to write a generalized main routine which would access the various parts of the algorithm and their variations on a subroutine call basis. In this way the different sub-algorithms could be coded and tested independently within the main framework . Figure 5.1 shows a functional flow diagram which depicts the format of the coding. Appendix I is a listing of the main routine and the subroutines used in the final algorithm. Parts of the code use system unique utility programs which were developed by and for the Magnavox Image Processing Facility and are contained in the system library. These subroutines are principally I/O routines and are not listed in the appendix. The algorithm code was written in a fortran 5 language supported by the Data General Eclipse computer.

Figure 5.1. Logical Flow Chart for Algorithm
(As Coded Into Main Routine)



6.0 CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

A good detection algorithm must have a high sensitivity, giving a high detection rate, but it must also have a high rejectivity, giving a low false alarm rate. The high sensitivity requires the ability to detect low wire signals in the presence of high noise and clutter levels which corrupt the signal. This means that it must in some way average out the noise and clutter fluctuations. As can be seen from the video tape demonstration accompanying this report, the semi-linear detection algorithm developed here has done fairly well in this area. The high rejectivity requires the ability to discriminate against edges and points as well as against line segments that do not constitute extended lines. The semi-linear detector algorithm does quite well on discriminating against edges and points, while the semi-local line discriminator algorithm does a remarkable job on eliminating isolated line segments (line shaped noise or clutter).

Yet it appears that the algorithm is still no substitute for the human visual system. It is evident from our tests that the detectivity and rejectivity of the human eye/brain system when interrogating a video scene is still superior to the above algorithm. In fact, the human video system may be close to the limit of performance that can be expected from the data, particularly in real time. Consider that the human eye has an extraordinarily wide dynamic range and a remarkable sensitivity in this range. It can detect differences in average brightness of as little as 2% over a dynamic range of nearly 1,000 centered about the eyes adapting brightness. This, taken together with the eye/brain ability to average over many pixels in a large neighborhood makes for a very sensitive detector. Furthermore, recent evidence exists that most of the cortical cells of the brain respond not to spots but to specifically oriented

line segments. Add to this the brain's remarkable ability to connect local detections in one area of a scene with local detections in another area, thereby giving an extraordinary degree of rejection of detected line segments that do not constitute a line (e.g., clutter), and one has an extremely efficient line extracting system.

Nevertheless, the algorithm developed here does quite a creditable job and it may be superior than the eye/brain system when the brain is preoccupied with a variety of other distracting tasks associated with flying, target acquisition and fire control, etc. The algorithm, as it stands, may be sufficient when combined with the eye/brain system so that the cable enhanced video gives the pilot sufficient warning to avoid most cables.

On the other hand, there is still considerable room for improvement, and there are two areas in which this improvement could be accomplished. These are:

a. Enlarge the local neighborhood sampling to improve the noise filtering effects of the averaging. This area certainly would increase the computation time, but should also result in a more sensitive detection mechanism.

b. Develop a global connection algorithm which would eliminate the residual clutter noise of the semi-local line discriminator. Again this would increase computation time but it would give a degree of rejection equivalent to the eye. It may be possible here to include additional features for discrimination such as the constraints of hanging cables. This could lead to an algorithm which discriminates against lines that are not caused by hanging wires. Remembering that every increase in discrimination ability allows a further reduction in the optimum detection

thresholds, we see that this global line connection algorithm would lead to increased sensitivity of the overall system.

As noted, these improvements do not come free. They will require increased memory capacity and increased computation time. The current version of the algorithm, when efficiently coded and implemented in hardware should be able to operate in real time. Only a thorough investigation will determine the degree to which these algorithms can be improved within the real-time and light weight constraints established by the mission requirements.

6.2 RECOMMENDATIONS

6.2.1 MAGIEC feels that while the results of the algorithms developed here are very encouraging and may be useful as they stand, it is premature to enter a hardware implementation program at this point. MAGIEC recommends that the hardware development effort be scheduled to commence at the end of a second phase of software development. We feel that significant improvement in performance and utility may be gained with an aggressive software improvement program. The level of effort expended to date simply was not sufficient to develop such a complex algorithm as may be necessary to approach the performance of the human visual system. This software improvement program should be aimed at those areas indicated above, namely:

a. Enlarging the neighborhood sampling of the detection algorithm to increase sensitivity.

b. Develop a global connection algorithm to discard line-like clutter.

We believe these objectives are realizable, but they are sizeable and will require a substantial level of effort to obtain them.

Once a global connection algorithm has been developed, it may be possible in the future to develop an alarm system which can make decisions, alert the pilot to potential danger, and recommend an evasive course.

APPENDIX 1

```
C****THIS IS THE MAIN ROUTINE USING SAMSET1.LINOP2.LINTES4.THRES3
      DIMENSION NAME(5).JJO(9.0:255).MED(5).INAME(5)
      COMMON/RED/JBLK,LINE,JO(0:255),JIR(4096)
      COMMON/RIT/NBLK,LIN.KO(0:255),IJQ(4096)
      COMMON/ANGLE/KA(0:255)
      COMMON/WINDOW/J9(9,9).JA(3).JB(3).JC(3)
      COMMON/LINE/JVAL(8)
      JBLK=0
      NBLK=0
      LINE=0
      LIN=0
      DO 8 I=0.255
    8 KO(I)=KA(I)=0
      TYPE"ENTER READ FILENAME"
      READ(11.100)NAME
  100 FORMAT(5A2)
      OPEN 2.NAME.ATT="C",REC=128
      OPEN 3,"TEMP.DC".ATT="C".REC=256
      DO 9 I=0.3
    9 CALL BRITE
      TYPE"ENTER THRESHOLD LINE STRENGTH  "
      READ FREE(11)ITHRES
      WRITE(10.50)ITHRES
   50 FORMAT("STRENGTH=".I5)
C****READ FIRST EIGHT ROWS******
      DO 1 J=2.9
      CALL READ1
      DO 1 JJ=0.255
 1    JJO(J.JJ)=JO(JJ)
C****DO THE WHOLE PICTURE******
      DO 2 KROW=4.251
C*****STORE NINE ROWS FOR THE SAMPLING MATRIX*****
      DO 3 K=1.8
      DO 3 J=0,255
   3  JJO(K.J)=JJO(K+1,J)
      CALL READ1
      DO 4 J=0.255
    4 JJO(9.J)=JO(J)
C****COMPUTE OPERATOR FOR THIS ROW*******
      DO 5 J=4.251
C****COMPUTE OPERATOR FOR  THIS PIXEL****
      DO 6 K=1.9
      DO 6 L=1.9
    6 J9(K.L)=JJO(K,J+L-5)
      DO 7 I=1.8
```

```
      CALL SAMSET1(I)
    7 CALL LINOP2(JVAL(I))
      CALL LINTES1(LINVAL,IVAL)
      IF(IABS(LINVAL).GT.0)CALL THRES1(ITHRES,LINVAL,IVAL)
      KA(J)=IVAL
    5 KO(J)=LINVAL
      CALL DIRECT(KROW)
    2 CONTINUE
      DO 10 I=251,255
   10 CALL BRITE
      CALL RESET
      TYPE '<7>                                        <7>'
      JBLK=0
      NBLK=0
      LINE=0
      LIN=0
      OPEN 2,'TEMP.DC',ATT='C',REC=256
      OPEN 3, NAME,ATT='C',REC=128
      DO 11 I=0,255
      CALL READ2
      CALL BREAD
      DO 12 J=6,249
      IF(JO(J)) 14,15,16
   14 KO(J)=0
   15 GO TO 12
   16 KO(J)=255
   12 CONTINUE
      CALL QANT1(I,1,KO)
   11 CONTINUE
      CALL RESET
      TYPE '<7>                                                    <7>'
      END
```

```
      SUBROUTINE SAMSET1(N)
C***9X9 SAMPLING CONFIGURATION,EIGHT ORIENTATIONS
      COMMON/WINDOW/J9(9,9),JA(3),JB(3),JC(3)
      JB(2)=J9(5,5)
      GO TO (10,20,30,40,50,60,70,80) N
   10 JB(1)=J9(5,3)
      JB(3)=J9(5,7)
      JA(1)=J9(3,3)
      JA(2)=J9(3,5)
      JA(3)=J9(3,7)
      JC(1)=J9(7,3)
      JC(2)=J9(7,5)
      JC(3)=J9(7,7)
      GO TO 90
   20 JB(1)=J9(6,3)
      JB(3)=J9(4,7)
      JA(1)=J9(4,2)
      JA(2)=J9(3,4)
      JA(3)=J9(2,6)
      JC(1)=J9(8,4)
      JC(2)=J9(7,6)
      JC(3)=J9(6,8)
      GO TO 90
   30 JB(1)=J9(7,3)
      JB(3)=J9(3,7)
      JA(1)=J9(5,1)
      JA(2)=J9(3,3)
      JA(3)=J9(1,5)
      JC(1)=J9(9,5)
      JC(2)=J9(7,7)
      JC(3)=J9(5,9)
      GO TO 90
   40 JB(1)=J9(7,4)
      JB(3)=J9(3,6)
      JA(1)=J9(6,2)
      JA(2)=J9(4,3)
      JA(3)=J9(2,4)
      JC(1)=J9(8,6)
      JC(2)=J9(6,7)
      JC(3)=J9(4,8)
      GO TO 90
   50 JB(1)=J9(7,5)
      JB(3)=J9(3,5)
      JA(1)=J9(7,3)
      JA(2)=J9(5,3)
```

```
      JA(3)=J9(3,3)
      JC(1)=J9(7,7)
      JC(2)=J9(5,7)
      JC(3)=J9(3,7)
      GO TO 90
   60 JB(1)=J9(7,6)
      JB(3)=J9(3,4)
      JA(1)=J9(8,4)
      JA(2)=J9(6,3)
      JA(3)=J9(4,2)
      JC(1)=J9(6,8)
      JC(2)=J9(4,7)
      JC(3)=J9(2,6)
      GO TO 90
   70 JB(1)=J9(7,7)
      JB(3)=J9(3,3)
      JA(1)=J9(9,5)
      JA(2)=J9(7,3)
      JA(3)=J9(5,1)
      JC(1)=J9(5,9)
      JC(2)=J9(3,7)
      JC(3)=J9(1,5)
      GO TO 90
   80 JB(1)=J9(6,7)
      JB(3)=J9(4,3)
      JA(1)=J9(4,8)
      JA(2)=J9(3,6)
      JA(3)=J9(2,4)
      JC(1)=J9(8,6)
      JC(2)=J9(7,4)
      JC(3)=J9(6,2)
   90 RETURN
      END
```

```
      SUBROUTINE LINOP2(JVAL)
C***THIS DEFINES THE LINE CHARACTERISTIC AND/OR OPERATOR TO BE USED
      COMMON/WINDOW/J9(9,9),JA(3),JB(3),JC(3)
      JASUM=JA(1)+JA(2)+JA(3)
      JBSUM=JB(1)+JB(2)+JB(3)
      JCSUM=JC(1)+JC(2)+JC(3)
      JVAL=0
      JBA=JBSUM-JASUM
      JBC=JBSUM-JCSUM
      IF(ISIGN(1,JBA).EQ.ISIGN(1,JBC))JVAL=JBA+JBC
      RETURN
      END
```

```
      SUBROUTINE LINTES1(LINVAL,IVAL)
C***THIS COMPARES ORIENTATIONS AND CHOOSES THAT WHICH HAS MAXIMUM EVIDENCE
      COMMON/LINE/JVAL(8)
      IVAL=0
      LINVAL=0
      DO 10 I=1,8
      JV=IABS(JVAL(I))
      IF(JV.GT.LINVAL)IVAL=I
   10 IF(JV.GT.LINVAL)LINVAL=JV
      LINVAL=JVAL(IVAL)
      IVAL=ISIGN(IVAL,LINVAL)
      RETURN
      END
```

```
SUBROUTINE THRES1(ITHRES,LINVAL,IVAL)
IF(IABS(LINVAL).LT.ITHRES)IVAL=0
IF(IABS(LINVAL).LT.ITHRES)LINVAL=0
RETURN
END
```

```
      SUBROUTINE DIRECT(KROW)
      COMMON/RIT/NBLK,LIN,KO(0:255),IJQ(4096)
      COMMON/ANGLE/KA(0:255)
      COMMON/VALUE/KVAL(5,0:255),KANG(5,0:255)
      IF(KROW.LE.7)GO TO 4
      DO 1 J=0,255
      DO 2 K=1,4
      KANG(K,J)=KANG(K+1,J)
    2 KVAL(K,J)=KVAL(K+1,J)
      KANG(5,J)=KA(J)
    1 KVAL(5,J)=KO(J)
      DO 3 J=6,249
      KO(J)=KVAL(3,J)
      IF(KO(J).EQ.0)GO TO 3
      CALL CLUST(J,JFLAG,NL)
      IF(NL.LE.2)KO(J)=0
      IF(JFLAG.NE.2)KO(J)=0
    3 CONTINUE
      DO 9 J=0,5
   9  KO(J)=0
      DO 10 J=250,255
  10  KO(J)=0
      GO TO 6
    4 DO 5 J=0,255
      KANG(KROW-2,J)=KA(J)
      KVAL(KROW-2,J)=KO(J)
    5 KA(J)=KO(J)=0
    6 IF(KROW.GT.5)CALL BRITE
      IF(KROW.LT.251)GO TO 7
      DO 8 J=0,255
    8 KO(J)=0
      CALL BRITE
    7 RETURN
      END
```

```
      SUBROUTINE CLUST(J,JFLAG,NL)
      COMMON/VALUE/KVAL(5,0:255),KANG(5,0:255)
      JFLAG=0
      NL=0
      K=IABS(KANG(3,J))
      GO TO (10, 20, 30, 40, 50, 60, 70, 80) K
   10 KAP=KANG(3,J+2)
      KAM=KANG(3,J-2)
      GO TO 90
   20 KAP=KANG(2,J+2)
      KAM=KANG(4,J-2)
      GO TO 90
   30 KAP=KANG(1,J+2)
      KAM=KANG(5,J-2)
      GO TO 90
   40 KAP=KANG(1,J+1)
      KAM=KANG(5,J-1)
      GO TO 90
   50 KAP=KANG(1,J)
      KAM=KANG(5,J)
      GO TO 90
   60 KAP=KANG(1,J-1)
      KAM=KANG(5,J+1)
      GO TO 90
   70 KAP=KANG(1,J-2)
      KAM=KANG(5,J+2)
      GO TO 90
   80 KAP=KANG(2,J-2)
      KAM=KANG(4,J+2)
   90 IF(KAP.EQ.KANG(3,J)) JFLAG=JFLAG+1
      IF(KAM.EQ.KANG(3,J)) JFLAG=JFLAG+1
      DO 1 K=-1,1
      DO 1 L=-1,1
    1 IF(KANG(3+K,J+L).EQ.KANG(3,J))NL=NL+1
      RETURN
      END
```

END